

Review on Public Cloud topics

EE4221

Lab 3

- Google App Engine Datastore is SQL database service.
 - FALSE
- Google App Engine can be used for hosting Web applications.
 - TRUE
- Performing a query involves two Datastore Python API methods : query and fetch.
 - TRUE
- Google App Engine developer server is run on Google Cloud.
 - FALSE
- webapp2 is a simple Web application framework.
 - TRUE

Lab 3

- webapp2 framework executes GAE Datastore queries.
 - FALSE
- GAE developer must implement user sessions management based on users' Google account username.
 - FALSE
- You can monitor your GAE applications on the Google's Web console.
 - TRUE
- To enable efficient queries, we need to define which property values will be used in the queries.
 - TRUE

Lab 4

- The following key-value pairs are valid results of running Word Count mapreduce job:
<cloud,10>, <computing,0>
 - FALSE
- Shuffling stage is done automatically by GAE Python mapreduce library.
 - TRUE
- A map function produces a list of intermediate keys.
 - TRUE
- *yield* method is used to pass computation results to GAE Python mapreduce library.
 - TRUE
- The programmer needs to specify how many concurrent workers is used for running GAE mapreduce job.
 - TRUE

Lab 4

- Mapreduce input files are stored in GAE Datastore.
 - FALSE
- Mapreduce input files are stored as blobs in GAE Blobstore.
 - TRUE
- Reduce function in Word Count can be called many time for each word.
 - FALSE
- The following is a valid output of content count mapreduce (counting the number of words in each file): <file1,5>, <file2, 502>, <file3,0>
 - FALSE
- Content count mapreduce (counting the number of words in each file) uses file name as a key and word count as a value.
 - TRUE

Test 2

- Web indexing is the main technique adopted by most search engines.
 - TRUE
- GAE Blobstore is a file storage.
 - TRUE
- Hybrid cloud combines private and public cloud.
 - TRUE
- Cloud providers need to follow predefined business models.
 - FALSE
- The user must create unique blob keys when uploading files in Blobstore.
 - FALSE
- Latin Pig is a high-level textual language used e.g. for defining Mapreduce jobs.
 - TRUE

Test 2

- When using MapReduce (e.g. with GAE) the programmer needs to handle issues of parallel and distributed programming.
 - FALSE
- SQL databases are based on predefined schema.
 - TRUE
- SQL databases are good solution for unstructured data.
 - FALSE
- Hadoop is a platform for parallel data processing.
 - TRUE
- Big Data is usually well structured and large volume.
 - FALSE
- Amazon Web Services provided IaaS and PaaS level services.
 - TRUE

Test 2

- Amazon Web Services can be used via programmatic interfaces (APIs).
 - TRUE
- Mapreduce can be used for data transformation and data aggregation.
 - TRUE
- 'Velocity' is one of the characteristics of big data.
 - TRUE
- In Mapreduce, Reduce function is called only once for each unique key value.
 - TRUE
- Mapreduce is designed such that several instances of map and reduce functions can be run in parallel.
 - TRUE
- Mapreduce is designed such that map and reduce functions can be executed on several virtual machines.
 - TRUE

4.1 ContentCount

- In which file(s) each word occurs
- Input: [file1: hello world map reduce] [file2: hello cloud computing] [file3: hello cloud]
- Final result: <hello, <file1, file2, file3> >, <world, <file1>>...

Lab 4.1 ContentCount

```
def content_map(data):
    """Index demo map function."""
    (entry, text_fn) = data
    text = text_fn()

    for s in split_into_sentences(text):
        for w in split_into_words(s.lower()):
            yield (w, entry.filename)

def content_reduce(key, values):
    """Index demo reduce function."""
    yield "%s: %s\n" % (key, list(set(values)))
```

4.2 ContentCount

- Count how many words each file contains
- Input: [file1: hello world map reduce] [file2: hello cloud computing] [file3: hello cloud]
- Final result: <file1, 4>, <file2, 3>, <file3, 2>
- Output format should be list of key-value pairs (string, digit) in the following format
 - input/file1.txt: 6
 - input/file2.txt: 18
 - input/file3.txt: 20

Lab 4.2 ContentCount

```
"""To be implemented by you"""  
def content_map(data):  
    """Content Count map function."""  
    (entry, text_fn) = data  
    text = text_fn()  
  
    for s in split_into_sentences(text):  
        for w in split_into_words(s.lower()):  
            yield ( entry.filename, w)
```

```
"""To be implemented by you"""  
def content_reduce(key, values):  
    """Content Count reduce function."""  
    yield "%s: %d\n" % (key, len(values))
```